Sign in

Google

Web Images Video News Maps **more »**

factored assert chain Search Advanced Search
Preferences

Web Results **1** - **10** of about **110,000** for **factored assert chain**. (**0.33** seconds)

Creating Secure Web Service Sessions @ SOA WORLD MAGAZINE
Standards such as WS-Security, the Security **Assertion** Markup Language (SAML), ... 2) Send the signed SAML **assertion** from the Web Service consumer to the Web ...
webservices.sys-con.com/read/250516.htm - 80k - Cached - Similar pages

Creating Secure Web Service Sessions @ SOA WORLD MAGAZINE
Standards such as WS-Security, the Security **Assertion** Markup Language (SAML), ... the concept of a "Web Service session" should be **factored** in. ...
webservices.sys-con.com/read/250516_1.htm - 79k - Cached - Similar pages

[PDF] Creating Secure Web Service Sessions
File Format: PDF/Adobe Acrobat - View as HTML
sense to re-**factor** your Web Services security solution in these situations. ... 2) Send the signed SAML **assertion** from the Web Service consumer to the Web ...
www.mcdonaldbradley.com/intPages/2006_Articles/
Creating_secure_webservices_ksmith_0706.pdf - Similar pages

[PDF] 1999 - Mathematical Forms and Strategies
File Format: PDF/Adobe Acrobat - View as HTML
Equations without variables **assert** the equality of different forms of the same number. ... But, if they do they can be **factored** and placed in Form 2, ...
me.nmsu.edu/~aseemath/3365_99_2.PDF - Similar pages

Creating Secure Web Service Sessions @ SOA WORLD MAGAZINE
2) Send the signed SAML **assertion** from the Web Service consumer to the Web Service in the WS-Security ... SOA Manufactures Success for the Supply **Chain** ...
soa.sys-con.com/read/250516.htm - 79k - Cached - Similar pages

JSTOR: On Monomial Algebras of Finite Global Dimension
Any 4-prechain w would require a 3-**chain** as a left **factor**, ... To prove this **assertion**, let al -.. am be the atomic factorization of any proper left segment ...
links.jstor.org/sici?sici=0002-9947(198509)291%3A1%3C291%3AOMAOFG%3E2.0.CO%3B2-T - Similar pages

JSTOR: A Performance Analysis of a Simple Prime-Testing Algorithm
However, if the program has gone down and we **factor** up to B -- 106 on one of the lower level numbers in the recursive **chain**, we stop the program. ...
links.jstor.org/sici?sici=0025-5718(198304)40%3A162%3C709%3AAPAOAS%3E2.0.CO%3B2-J - Similar pages
[More results from links.jstor.org]

[PDF] Multivariate, Replicated, Single-Subject, Repeated Measures ...
File Format: PDF/Adobe Acrobat
using **chain** P-technique **factor** analysis will be mentioned ... We **assert** that the power of MRSRM designs. and P-technique **factor** analysis will not become ...
www.blackwell-synergy.com/doi/pdf/10.1111/j.1741-2358.1990.tb00273.x - Similar pages

[PS] Markov **Chain** Monte Carlo Convergence Diagnostics: A Comparative Review
File Format: Adobe PostScript - View as Text
chain variance and the within-**chain** variance. Convergence is monitored by estimating the **factor** by. which the scale parameter might shrink if sampling were ...
ftp://muskie.biostat.umn.edu/pub/1994/rr94-008.ps.Z - Similar pages

[PDF] Extending Continuous Time Bayesian Networks

File Format: PDF/Adobe Acrobat - View as HTML
negative evidence, by introducing updates that **assert** no ob- ... Also like other **factored** representations such as Bayesian ...
www.cs.washington.edu/homes/weld/papers/aaai05_CTBN.pdf - Similar pages

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

factored assert chain Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Sign in

Google

Web Images Video News Maps **more »**

information variable program Search | Advanced Search
Preferences

Web

Results **1** - **10** of about **108,000,000** for **information variable program**. (**0.21** seconds)

CGI Environment **Variables**
In order to pass data about the **information** request from the server to the script, the server uses command line arguments as well as environment **variables**. ...
hoohoo.ncsa.uiuc.edu/cgi/env.html - 6k - Cached - Similar pages

Student Aid on the Web
Both the FFEL and Direct Loan **programs** consist of what are generally known as ... a borrower's **variable** interest rate on loans made before July 1, 2006. ...
www.studentaid.ed.gov/PORTALSWebApp/students/english/studentloans.jsp - 32k - Cached - Similar pages

variable: **Information** from Answers.com
The collection of results from an experiment, or **information** to be used to draw ... The scope of a **variable** is the portion of the **program** code for which the ...
www.answers.com/topic/**variable**-1 - 73k - Cached - Similar pages

AAVSO: Observing **Programs** of the AAVSO
Special finder charts for each star are available, with the **variable**, a comparison star, and a check star indicated. For more **information** on the PEP **program** ...
www.aavso.org/observing/**programs**/ - 22k - Cached - Similar pages

FreeFlow Makeready 5.0 and More Workflow Software from Xerox
Set labeling feature easily adds **variable** content to documents such as covers, ... For general **information** about compatibility and performance or to order ...
www.xerox.com/digital-printing/workflow/FreeFlow/MakeReady/enus.html - 62k - Cached - Similar pages

Xerox Store - **Variable Information** Software
Creative Resources · Document and **Information** Capture · ProfitAccelerator™ Resources · **Variable Information** Software ...
www.graphiccommunications-estore.xerox.com/commerce/Product.aspx?MSCSProfile=DCCDF22EB27065BEEB660F3E97ED... - 41k - Cached - Similar pages

Emco **variables** software by EMCO and others
... allowing them to exchange **information** in real-time. Simply share **variables** among all your VB.NET, VB6 and C **programs** as if they where global **variables**. ...
www.freedownloadscenter.com/Best/emco-**variables**.html - 34k - Cached - Similar pages

Variables Software
A VB **program** can use any number of beWise **Variables** simultaneously, ... script execution time, the page source and other **information** that helps you in your ...
www.**program**url.com/software/**variables**.htm - 80k - Cached - Similar pages

Custom **Variables** QuestionPro Online Survey Software
Custom **Variables** are used to store additional **information** that is to be passed to ... How can I use the ext_ref **variable** with my own email **program**/provider? ...
www.questionpro.com/help/131-window.html - 15k - Cached - Similar pages

Session **variables** software
Related keywords: session, **variables**, Session server, User session **information**, application **variables**, Session, session log, environment **variables**, ...
www.allworldsoft.com/keywords/session_**variables**/ - Similar pages

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

information variable program Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Sign in

Google

Web Images Video News Maps **more »**

assert statement program | Search | Advanced Search
Preferences

Web

Results **1 - 10** of about **1,120,000** for **assert statement program**. (**0.36** seconds)

Programming With Assertions
The second form of the **assertion statement** should be used in preference to the first only when the **program** has some additional information that might help **...**
java.sun.com/j2se/1.4.2/docs/guide/lang/**assert**.html - 42k - Cached - Similar pages

[PDF] IBM **Statement** of Non-**Assertion** of Named Patents Against OSS
File Format: PDF/Adobe Acrobat - View as HTML
create innovative software **programs**, the commitment not to **assert** any of these **...** Method and system for controlling the generation of **program statements ...**
www.ibm.com/ibm/licensing/patents/pledgedpatents.pdf - Similar pages

Using Testability to Guide **Assertion** Placement
When the user places an **ASSERT statement** using the AEW, it is not immediately placed in the **program's** source code. Assertions built in this manner are first **...**
hissa.nist.gov/latex/node25.html - 8k - Cached - Similar pages

[PPT] The **assert statement**
File Format: Microsoft Powerpoint - View as HTML
The purpose of the **assert statement** is to give you a way to catch **program** errors early; The **assert statement** is new in Java 1.4; The **assert statement** uses a **...**
www.cis.upenn.edu/~matuszek/cit591-2004/Lectures/21-**assert**.ppt - Similar pages

Javanotes 5.0, Section 8.4 -- Assertions
If a **program** is run with assertions disabled, an **assertion statement** is equivalent to an empty **statement** and has no effect. When assertions are enabled and **...**
math.hws.edu/javanotes/c8/s4.html - 14k - Cached - Similar pages

Assertion (computing) - Wikipedia, the free encyclopedia
The use of assertions helps the programmer design, develop, and reason about a **program**. The use of **assertion statements** provides additional help during **...**
en.wikipedia.org/wiki/**Assert**ion_(computing) - 32k - Cached - Similar pages

CppAssert at CodePedia
In above **program**, the function getFraction() is called to get a fraction. **...** Below is an example how to make your own **assert statement**. **...**
www.codepedia.com/1/Cpp**Assert** - 16k - Cached - Similar pages

The **assert** macro - HP DSPP
You can have all the debugging **assert statements** you like and it won't have any impact on the final, non-debug, **program**. The argument to an **assert statement ...**
h21007.www2.hp.com/dspp/tech/tech_TechSingleTipDetailPage_IDX/1,2366,3690,00.html - 43k - Cached - Similar pages

JML Reference Manual: **Statements** and Annotation **Statements**
An **assert statements** tells JML to check that the specified predicate is true at the given point in the **program**. The runtime **assertion** checker checks such **...**
www.cs.iastate.edu/~leavens/JML/jmlrefman/jmlrefman_12.html - 34k - Cached - Similar pages

remove **assert statement** (Was: Re: PEP new **assert** idiom)
That is, if the **program** is bug-free, this condition will never happen. An **assertion ...** The **assert statement** declares this to be an impossible condition. **...**
mail.python.org/pipermail/python-list/2004-November/291402.html - 9k - Cached - Similar pages

Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

assert statement program Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Sign in

Google

Web Images Video News Maps **more »**

assert chain java | Search | Advanced Search
Preferences

Web

Results **1** - **10** of about **348,000** for **assert chain java**. (**0.23** seconds)

Mailing list archives
getTargetInvoker(); + **assert chain** != null; + invoker = **chain**. ... +++ incubator/tuscany/sandbox/jboynes/sca/core2/src/test/**java**/org/apache/tuscany/core/ ...
mail-archives.apache.org/.../ws-tuscany-commits/200605.mbox/%3C20060522003153.3F2081A983A@eris.apache.org%3E - 19k - Cached - Similar pages

Mailing list archives
WorkContext; import static org.apache.tuscany.spi.idl.**java**. ... private transient String convIdForRemotableTarget; @@ -129,6 +135,11 @@ **assert chain** ! ...
mail-archives.apache.org/.../ws-tuscany-commits/200612.mbox/%3C20061212214202.BF34D1A981C@eris.apache.org%3E - 15k - Cached - Similar pages
[More results from mail-archives.apache.org]

fw at deneb dot enyo - [Bug **java**/16927] New: **assert** statement ...
StringBuffer tree_2>>> **chain** <tree_list 0x403db5b8 purpose <call_expr 0x403e2500 side-effects ... [Bug **java**/16927] **assert** statement fails at -O1 and above ...
gcc.gnu.org/ml/**java**-prs/2004-q3/msg00197.html - 9k - Cached - Similar pages

Bryce McKinlay - **Java**: [BC] Implement type **assertion** table
This might resolve classes in one **chain** or the other. ... Follow-Ups:. Re: **Java**: [BC] Implement type **assertion** table. From: Tom Tromey ...
gcc.gnu.org/ml/**java**-patches/2004-q4/msg00468.html - 36k - Cached - Similar pages
[More results from gcc.gnu.org]

Bug ID: 4356294 "**assert**" is no longer a legal **Java** identifier
java:compiler, "**assert**" is no longer a legal **Java** identifier,State: closed ... be used as an identifier Util.**assert**(target > **chain**.pc || stacksize == 0); . ...
bugs.sun.com/bugdatabase/view_bug.do?bug_id=4356294 - 27k - Cached - Similar pages

Borland Security Service: Class CertificateWallet
String alias, **java**.security.cert.X509Certificate[] **chain**, **java**.security. ... A wallet constructed using this constructor may only be used to **assert** an ...
info.borland.com/techpubs/bes/v6/html_books/sec-api-doc/com/borland/security/provider/CertificateWallet.html - 26k - Cached - Similar pages

Behavioral Specification of Distributed Software Component Interfaces
We developed a new package, **java**.lang.**assert**, to define five new exceptions ... The **assertion chain** method iteratively calls other **assertion chain** methods ...
doi.ieeecomputersociety.org/10.1109/2.774918 - Similar pages

Cookbook - Commons **Chain**
Assert; import org.apache.commons.**chain**.mailreader.MailReader; import **java**.util.Locale; public class LocaleValueTest extends TestCase { MailReader context; ...
jakarta.apache.org/commons/**chain**/cookbook.html - 61k - Cached - Similar pages

BEA AquaLogic Enterprise Security Provider SSPI 2.1 Reference ...
Used when a CSIv2 X509 certificate **chain** identity token is passed during an invoke. static **java**.lang.String, SAML_ASSERTION_TYPE Used when a SMAL **Assertion** ...
e-docs.bea.com/ales/docs21/**java**docs/SSPI/weblogic/security/spi/Identity**Assert**er.html - 16k - Cached - Similar pages

[PDF] Behavioral Specification of Distributed Software Component Interfaces
File Format: PDF/Adobe Acrobat
instance invariant as **java**.lang.**assert**.Invariant. Since. the current Biscotti implementation

does ... Boolean **assertion** is false. The **assertion chain** method ...
ieeexplore.ieee.org/iel5/2/16817/00774918.pdf?arnumber=774918 - Similar pages

Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

assert chain java | Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Sign in

Google

Web Images Video News Maps **more »**

assert denominator tree compiler | Search | Advanced Search
Preferences

Web

Results **1** - **10** of about **31,900** for **assert denominator tree compiler**. (**0.25** seconds)

[DOC] Procedural-
File Format: Microsoft Word - View as HTML
A syntax error is produced by a **compiler** when the parse **tree** cannot be **... assert denominator** not= 0. When a check fails at run time, an exception is raised **...**
ww2.cs.fsu.edu/~peress/PLangStudy.doc - Similar pages

Overview
Implemented in semantic analysis phase of the **compiler ... assert denominator** not= 0;
When a check fails at run time, an exception Define this term is **...**
www.cs.fsu.edu/~engelen/courses/COP402003/notes4.html - 25k - Cached - Similar pages

Semantic Analysis
assert denominator not= 0; When a check fails at run time, an exception **...** Semantic rules are used by a **compiler** to enforce static semantics and/or to **...**
www.cs.fsu.edu/~engelen/courses/COP402001/notes4.html - 23k - Cached - Similar pages
[More results from www.cs.fsu.edu]

Switches for GCC
Execute ver version of the **compiler**. This is the gcc version, not the XGC Ada version. **...**
This switch, where "a" stands for **assert**, causes **Assert** and Debug **...**
www.xgc.com/manuals/m1750-ada/xgc-ada-ug/x255.html - 67k - Cached - Similar pages

Switches for gcc
Execute ver version of the **compiler**. This is the gcc version, **...** This switch, where "a" stands for **assert**, causes **Assert** and Debug pragmas to be activated. **...**
www.xgc.com/manuals/erc32-ada-ug/node44.html - 74k - Cached - Similar pages

The GNU C Library - C Language Facilities Implemented By the Library
If NDEBUG is not defined, **assert** tests the value of expression . **...** In practice, the GNU C **compiler** always passes a given set of argument types in the same **...**
www.cs.utah.edu/dept/old/texinfo/glibc-manual-0.02/library_28.html - 46k -
Cached - Similar pages

CSci320: CS320 Prolog Examples 2
It helps to know that a fact we **assert** is not already in the data base. **...** with it does not mean understanding how the **compiler** stores the syntax **tree**! **...**
www.csci.csusb.edu/dick/cs320/lab/19.html - 33k - Cached - Similar pages

Andreas Jaeger - Results for 4.1.0 20050513 (experimental ...
... denominator: 1 1 FAIL: gcc.dg/**tree**-ssa/ltrans-5.c scan-**tree**-dump-times **...** 3 # of expected failures 23 **Compiler** version: 4.1.0 20050513 (experimental) **...**
gcc.gnu.org/ml/gcc-testresults/2005-05/msg00863.html - 18k - Cached - Similar pages

errata
Page 207: **Tree** should have arrows, not undirected lines Page 209: (old) **...** "(c " in rat+rat should be: (b (rat-**denominator** x)) (c (rat-numerator y)) Page **...**
www.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/lang/lisp/bookcode/norvig/errata.txt
- 19k - Cached - Similar pages

GameDev.net - Tile Graphics Techniques
I feel it's a fair **assertion** to assume that people who play games have at least 4 megs in their machine. Catering to the lowest common **denominator** (i.e., **...**
www.gamedev.net/reference/articles/article727.asp - 32k - Cached - Similar pages

Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

assert denominator tree compiler Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+assert +variable +usage SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used assert variable usage

Found **2,808** of **201,062**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200
Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Relevance scale □□■■■

1 A cost effective question asking strategy
Evangelos Triantaphyllou, Jinchang Wang
April 1992 **Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing: technological challenges of the 1990's SAC '92**
Publisher: ACM Press
Full text available: pdf(611.10 KB) Additional Information: full citation, references, index terms

2 Types with semantics: soundness proof assistant
Olha Shkaravska
September 2005 **Proceedings of the 3rd ACM SIGPLAN workshop on Mechanized reasoning about languages with variable binding MERLIN '05**
Publisher: ACM Press
Full text available: pdf(184.02 KB) Additional Information: full citation, abstract, references, index terms

We present a parametric Hoare-like logic for computer-aided reasoning about typeable properties of functional programs. The logic is based on the concept of a *specialised assertion*, which is a predicate expressing the semantics of a typing judgment in a logical framework (here higher-order logic). Replacing in a type inference rule the judgments by the appropriate specialised assertions, one obtains a *specialised rule*. Specialised assertions have a uniform format, and soundness pro ...

Keywords: assertion, automated theorem proving, program logic, type system

3 Control predicates are better than dummy variables for reasoning about program control
Leslie Lamport
April 1988 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 10 Issue 2
Publisher: ACM Press
Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms, review

When explicit control predicates rather than dummy variables are used, the Owicki-Gries method for proving safety properties of concurrent programs can be strengthened, making it easier to construct the required program annotations.

4 Optimizing sparse representations for dataflow analysis
Erik Ruf
March 1995 **ACM SIGPLAN Notices , Papers from the 1995 ACM SIGPLAN workshop**

on Intermediate representations, Volume 30 Issue 3
Publisher: ACM Press
Full text available: pdf(1.42 MB) Additional Information: full citation, abstract, citings, index terms

Sparse program representations allow inter-statement dependences to be represented explicitly, enabling dataflow analyzers to restrict the propagation of information to paths where it could potentially affect the dataflow solution. This paper describes the use of a single sparse program representation, the value dependence graph, in both general and analysis-specific contexts, and demonstrates its utility in reducing the cost of dataflow analysis. We find that several semantics-preserving t ...

5 Validation of Scientific Programs
William E. Howden
June 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 2
Publisher: ACM Press
Full text available: pdf(2.92 MB) Additional Information: full citation, references, citings, index terms

6 ASAP—a simple assertion pre-processor
Igor D.D. Curcio
December 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 12
Publisher: ACM Press
Full text available: pdf(803.78 KB) Additional Information: full citation, abstract, index terms

Assertions are widely known as a powerful tool to detect software faults during the debugging of software systems. Despite the maturity of software engineering tools, assertions are seldom used in practice. ASAP is a pre-processor for C programs which implements several concepts defmed in the theory of formal specification, such as preconditions, postconditions, assertions related to intermediate states, loop invariants and variants, existential and universal quantifiers. In this paper, the noti ...

Keywords: assertions, pre-processor, programming techniques, software contract, software engineering

7 Validating programs without specifications
W. Howden
November 1989 **ACM SIGSOFT Software Engineering Notes , Proceedings of the ACM SIGSOFT '89 third symposium on Software testing, analysis, and verification TAV3**, Volume 14 Issue 8
Publisher: ACM Press
Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, citings, index terms

This work was supported by the Office of Naval Research and the Naval Weapons Center

8 A software quality assurance experiment
J. P. Benson, S. H. Saib
January 1978 **ACM SIGSOFT Software Engineering Notes , ACM SIGMETRICS Performance Evaluation Review , Proceedings of the software quality assurance workshop on Functional and performance issues**, Volume 3 , 7 Issue 5 , 3-4
Publisher: ACM Press
Full text available: pdf(446.99 KB) Additional Information: full citation, abstract, references, citings, index terms

An experiment was performed to evaluate the ability of executable assertions to detect programming errors in a real time program. Errors selected from the categories of computational errors, data handling errors, and logical errors were inserted in the program. Assertions were then written which detected these errors. While computational errors were easily detected, data handling and logical errors were more difficult to locate. New types of assertions will be required to protect against th ...

Keywords: Assertions, Error categories

9 Software testing: Cutpoints for formal equivalence verification of embedded software
Xiushan Feng, Alan J. Hu
September 2005 **Proceedings of the 5th ACM international conference on Embedded software EMSOFT '05**
Publisher: ACM Press
Full text available: pdf(138.94 KB) Additional Information: full citation, abstract, references, citings, index terms

Like hardware, embedded software faces stringent design constraints, undergoes extremely aggressive optimization, and therefore has a similar need for verifying the functional equivalence of two versions of a design, e.g., before and after an optimization. The concept of cutpoints was a breakthrough in the formal equivalence verification of combinational circuits and is the key enabling technology behind its successful commercialization. We introduce an analogous idea for formally verifying the ...

Keywords: embedded software, equivalence checking, formal verification

10 Applicability of Software Validation Techniques to Scientific Programs
W. E. Howden
July 1980 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 2 Issue 3
Publisher: ACM Press
Full text available: pdf(914.46 KB) Additional Information: full citation, abstract, references, citings, index terms

Error analysis involves the examination of a collection of programs whose errors are known. Each error is analyzed and validation techniques which would discover the error are identified. The errors that were present in version five of a package of Fortran scientific subroutines and then later corrected in version six were analyzed. An integrated collection of static and dynamic analysis methods would have discovered the errors in version five before its release. An integrated approach to v ...

11 A logical analysis of aliasing in imperative higher-order functions
Martin Berger, Kohei Honda, Nobuko Yoshida
September 2005 **ACM SIGPLAN Notices , Proceedings of the tenth ACM SIGPLAN international conference on Functional programming ICFP '05**, Volume 40 Issue 9
Publisher: ACM Press
Full text available: pdf(234.59 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a compositional program logic for call-by-value imperative higher-order functions with general forms of aliasing, which can arise from the use of reference names as function parameters, return values, content of references and parts of data structures. The program logic extends our earlier logic for alias-free imperative higher-order functions with new modal operators which serve as building blocks for clean structural reasoning about programs and data structures in the presence of al ...

Keywords: π-calculus, aliasing, functional programming, hoare-logics, modalities, pointers, typing

12
Programming and verifying critical systems by means of the synchronous data-flow language LUSTRE

Christophe Ratel, Nicolas Halbwachs, Pascal Raymond
September 1991 **ACM SIGSOFT Software Engineering Notes , Proceedings of the conference on Software for citical systems SIGSOFT '91**, Volume 16 Issue 5
Publisher: ACM Press
Full text available: pdf(738.92 KB) Additional Information: full citation, references, citings, index terms

13 Automatic TCP buffer tuning
Jeffrey Semke, Jamshid Mahdavi, Matthew Mathis
October 1998 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '98**, Volume 28 Issue 4
Publisher: ACM Press
Full text available: pdf(1.10 MB) Additional Information: full citation, abstract, references, citings, index terms

With the growth of high performance networking, a single host may have simultaneous connections that vary in bandwidth by as many as six orders of magnitude. We identify requirements for an automatically-tuning TCP to achieve maximum throughput across all connections simultaneously within the resource limits of the sender. Our auto-tuning TCP implementation makes use of several existing technologies and adds dynamically adjusting socket buffers to achieve maximum transfer rates on each connectio ...

14 Extracting library-based object-oriented applications
Peter F. Sweeney, Frank Tip
November 2000 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 8th ACM SIGSOFT international symposium on Foundations of software engineering: twenty-first century applications SIGSOFT '00/FSE-8**, Volume 25 Issue 6
Publisher: ACM Press
Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

In an increasingly popular model of software distribution, software is developed in one computing environment and deployed in other environments by transfer over the internet. Extraction tools perform a static whole-program analysis to determine unused functionality in applications in order to reduce the time required to download applications. We have identified a number of scenarios where extraction tools require information beyond what can be inferred through static analysis: software distr ...

15 From process logic to program logic
Kohei Honda
September 2004 **ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN international conference on Functional programming ICFP '04**, Volume 39 Issue 9
Publisher: ACM Press
Full text available: pdf(208.11 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a process logic for the p -calculus with the linear/affine type discipline [6, 7, 31, 32, 33, 59, 60]. Built on the preceding studies on logics for programs and processes, simple systems of assertions are developed, capturing the classes of behaviours ranging from purely functional interactions to those with destructive update, local state and genericity. A central feature of the logic is representation of the behaviour of an environment as the dual of that of a process in an assertio ...

Keywords: π-calculus, duality, higher-order functions, hoare logic, mobile processes, types

16 Exploiting implicit parallelism of logic languages with the SAM
Giancarlo Succi
April 1992 **Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing: technological challenges of the 1990's SAC '92**
Publisher: ACM Press
Full text available: pdf(784.34 KB) Additional Information: full citation, references, citings, index terms

17 An architecture for voice dialog systems based on prolog-style theorem proving
Ronnie W. Smith, Alan W. Biermann, D. Richard Hipp
September 1995 **Computational Linguistics**, Volume 21 Issue 3
Publisher: MIT Press
Full text available: pdf(2.76 MB) Publisher Site Additional Information: full citation, abstract, references, citings

A pragmatic architecture for voice dialog machines aimed at the equipment repair problem has been implemented. This architecture exhibits a number of behaviors required for efficient human-machine dialog. These behaviors include:(1) problem solving to achieve a target goal(2) the ability to carry out subdialogs to achieve appropriate subgoals and to pass control arbitrarily from one subdialog to another(3) the use of a user model to enable useful verbal exchanges and to inhibit unnecessary ones(...

18 Session 1: Performance analysis based upon complete profiles
Joan Krone, William F. Ogden, Murali Sitaraman
November 2006 **Proceedings of the 2006 conference on Specification and verification of component-based systems SAVCBS '06**
Publisher: ACM Press
Full text available: pdf(116.69 KB) Additional Information: full citation, abstract, references, index terms

A system for engineering and verifying component-based software must include mechanisms for specifying abstractly not only the complete functionality of components but their exact performance as well. This paper introduces *profiles* as a first-class construct for complete, independent specification of performance in higher-level languages. Using profiles, a developer can select from an assortment of implementations for a particular functionality the one that best suits his needs with respe ...

Keywords: components, performance, reuse, software engineering, specification

19 Social networks, networking & virtual communities: HT06, tagging paper, taxonomy, Flickr, academic article, to read
Cameron Marlow, Mor Naaman, Danah Boyd, Marc Davis
August 2006 **Proceedings of the seventeenth conference on Hypertext and hypermedia HYPERTEXT '06**
Publisher: ACM Press
Full text available: pdf(339.73 KB) Additional Information: full citation, abstract, references, index terms

In recent years, *tagging systems* have become increasingly popular. These systems enable users to add keywords (i.e., "tags") to Internet resources (e.g., web pages, images, videos) without relying on a controlled vocabulary. Tagging systems have the potential to improve search, spam detection, reputation systems, and personal organization while introducing new modalities of social communication and opportunities for data mining. This potential is largely due to the social structure that u ...

Keywords: Flickr, categorization, classification, folksonomy, incentives, models, research, social networks, social software, tagging systems, tagsonomy, taxonomy

20 An efficient assertion checker for combinational properties

Gagan Hasteer, Anmol Mathur, Prithviraj Banerjee
June 1997 **Proceedings of the 34th annual conference on Design automation DAC '97**
Publisher: ACM Press

Full text available: pdf(241.80 KB) Additional Information: full citation, abstract, references, citings, index terms

Formally verifying properties of signals in a circuit hasseveral applications in an equivalence checking based formalverification flow.In a hierarchical design, functionalityis divided across blocks.This necessitates the useof constraints on input signals of a block to avoid falsenegatives.Validating such input constraints requires assertionchecking at the outputs of modules generatingthe constrained signals.In this paper, we present anefficient assertion checker for combinational propertieswhic ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used **assert denominator tree**

Found **152** of **201,062**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 152 Result page: **1** 2 3 4 5 6 7 8 next

Relevance scale □ ▂ ▄ ▆ ■

1 Self-assessment procedure VIII: a self-assessment procedure dealing with the programming language Ada
Peter Wegner
October 1981 **Communications of the ACM**, Volume 24 Issue 10
Publisher: ACM Press
Full text available: pdf(2.41 MB)

January 1982 Book

Publisher: Addison-Wesley Longman Publishing Co., Inc.

Full text available: pdf(19.47 MB) Additional Information: full citation, abstract, references, citings, index terms

From the Preface (See Front Matter for full Preface)

Electronic computers have evolved from exiguous experimental enterprises in the 1940s to prolific practical data processing systems in the 1980s. As we have come to rely on these systems to process and store data, we have also come to wonder about their ability to protect valuable data.

Data security is the science and study of methods of protecting data in computer and communication systems from unauthorized disclosure ...

5 Some aspects of symbolic integration via predicate logic programming

Henry Kanoui

November 1976 **ACM SIGSAM Bulletin**, Volume 10 Issue 4

Publisher: ACM Press

Full text available: pdf(988.48 KB) Additional Information: full citation, abstract, references, citings

During the past years, various algebraic manipulations systems have been described in the literature. Most of them are implemented via "classic" programming languages like Fortran, Lisp, PL1 ... We propose an alternative approach: the use of Predicate Logic as a programming language.

6 DBMS implementation experience: A generalized DBMS implementation on a database machine

Asuman Dogac, Esen A. Ozkarahan

May 1980 **Proceedings of the 1980 ACM SIGMOD international conference on Management of data SIGMOD '80**

Publisher: ACM Press

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings

The design and implementation of a generalized database management system based on the RAP database machine is described. The GDBMS architecture, while is similar to the ANSI/SPARC view, has differences in the interpretation of its physical levels. The E/R model is chosen as the Meta Data Model (Conceptual Schema) which generates external model interfaces consisting of the relational, network, and hierarchical models. The SEQUEL, LSL, MRI --- like languages are supported at these interfaces. Str ...

Keywords: DDL, DML, E/R model query languages, GDBMS, RAP database machine, associative processors, data models, database machines, operational transformations, schemas, structural transformations

7 Generating editing environments based on relations and attributes

Susan Horwitz, Tim Teitelbaum

August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

Publisher: ACM Press

Full text available: pdf(2.38 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The ability to generate language-based editors depends on the existence of a powerful, language-independent model of editing. A model is proposed in which programs are represented as attributed abstract-syntax trees with an associated relational database. Relations can depend on the state of the attributed tree, and attributes can depend on the values in relations, provided there are no circular dependencies. The power and the limitations of relational operations are demonstrated ...

8 Learning classifiers: Using urls and table layout for web classification tasks
L. K. Shih, D. R. Karger
May 2004 **Proceedings of the 13th international conference on World Wide Web WWW '04**
Publisher: ACM Press
Full text available: pdf(357.43 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose new features and algorithms for automating Web-page classification tasks such as content recommendation and ad blocking. We show that the automated classification of Web pages can be much improved if, instead of looking at their textual content, we consider each links's URL and the visual placement of those links on a referring page. These features are unusual: rather than being scalar measurements like word counts they are *tree structured*---describing the position of the item ...

Keywords: classification, news recommendation, tree structures, web applications

9 Relations and attributes: A symbiotic basis for editing environments
Susan Horwitz, Tim Teitelbaum
June 1983 **ACM SIGPLAN Notices , ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments**, Volume 18 , 20 Issue 6 , 7
Publisher: ACM Press
Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, references, citings, index terms

The ability to generate language-based editors depends on the existence of a powerful, language-independent model of editing. A model is proposed in which programs are represented as attributed abstract-syntax trees with an associated relational database. Relations can depend on the state of the attributed tree, and attributes can depend on the values in relations, provided there are no circular dependencies. The power and the limitations of relational operations are demonstrated ...

10 Lower bounds for noisy Boolean decision trees
William Evans, Nicholas Pippenger
July 1996 **Proceedings of the twenty-eighth annual ACM symposium on Theory of computing STOC '96**
Publisher: ACM Press
Full text available: pdf(734.35 KB) Additional Information: full citation, references, citings, index terms

11 Computation: finite and infinite machines
Marvin L. Minsky
January 1967 Book
Publisher: Prentice-Hall, Inc.
Additional Information: full citation, abstract, references, citings, index terms

From the Preface (See Front Matter for full Preface)

Man has within a single generation found himself sharing the world with a strange new species: the computers and computer-like machines. Neither history, nor philosophy, nor common sense will tell us how these machines will affect us, for they do not do "work" as did machines of the Industrial Revolution. Instead of dealing with materials or energy, we are told that they handle "control" and "information" and even "intellectua ...

12 A lower bound for integer greatest common divisor computations
Yishay Mansour, Baruch Schieber, Prasoon Tiwari
April 1991 **Journal of the ACM (JACM)**, Volume 38 Issue 2

Publisher: ACM Press

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, index terms, review

It is proved that no finite computation tree with operations { +, -, *, /, mod, < } can decide whether the greatest common divisor (gcd) of a and b is one, for all pairs of integers a and b. This settles a problem posed by Gro¨tschel et al. Moreover, if the constants explicitly involved in any operation performed in the tree are restricted to be "0" and "1" (and any other constant must be comp ...

Keywords: floor operation, greatest common devisor, lower bound, mod operation, truncation

13 Design: a financial modelling system
Fred Appleyard, Roger Hui
May 1985 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL: APL and the future APL '85**, Volume 15 Issue 4
Publisher: ACM Press

Full text available: pdf(951.46 KB) Additional Information: full citation, abstract, references, citings, index terms

A non-procedural financial modelling system is presented. The financial models can be deterministic or probabilistic. Some issues in the design and implementation of this system are discussed.

14 Probabilistic inductive inference
L. Pitt
April 1989 **Journal of the ACM (JACM)**, Volume 36 Issue 2
Publisher: ACM Press

Full text available: pdf(4.04 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Inductive inference machines construct programs for total recursive functions given only example values of the functions. Probabilistic inductive inference machines are defined, and for various criteria of successful inference, it is asked whether a probabilistic inductive inference machine can infer larger classes of functions if the inference criterion is relaxed to allow inference with probability at least p, (0 < p < 1) as opposed ...

15 Alternating-time temporal logic
Rajeev Alur, Thomas A. Henzinger, Orna Kupferman
September 2002 **Journal of the ACM (JACM)**, Volume 49 Issue 5
Publisher: ACM Press

Full text available: pdf(345.26 KB) Additional Information: full citation, abstract, references, citings, index terms

Temporal logic comes in two varieties: *linear-time temporal logic* assumes implicit universal quantification over all paths that are generated by the execution of a system; *branching-time temporal logic* allows explicit existential and universal quantification over all paths. We introduce a third, more general variety of temporal logic: *alternating-time temporal logic* offers selective quantification over those paths that are possible outcomes of games, such as the game in whic ...

Keywords: Alternation, games, model checking, temporaxl logic

16 Efficient Detection of Network Motifs
Sebastian Wernicke
October 2006 **IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)**, Volume 3 Issue 4
Publisher: IEEE Computer Society Press

Full text available: pdf(2.52 MB) Additional Information: full citation, abstract, references, index terms

Motifs in a given network are small connected subnetworks that occur in significantly higher frequencies than would be expected in random networks. They have recently gathered much attention as a concept to uncover structural design principles of complex networks. Kashtan et al. [Bioinformatics, 2004] proposed a sampling algorithm for performing the computationally challenging task of detecting network motifs. However, among other drawbacks, this algorithm suffers from a sampling bias and scales ...

Keywords: Network motif detection algorithm, subgraph enumeration, subgraph sampling, subgraph concentration in random graphs.

17 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**
Publisher: IBM Press
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

18 A cost-benefit decision model: analysis, comparison amd selection of data management
Stanley Y. W. Su, Jozo Dujmovic, D. S. Batory, S. B. Navathe, Richard Elnicki
September 1987 **ACM Transactions on Database Systems (TODS)**, Volume 12 Issue 3
Publisher: ACM Press
Full text available: pdf(3.29 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a general cost-benefit decision model that is applicable to the evaluation, comparison, and selection of alternative products with a multiplicity of features, such as complex computer systems. The application of this model is explained and illustrated using the selection of data management systems as an example. The model has the following features: (1) it is mathematically based on an extended continuous logic and a theory of complex criteria; (2) the decisi ...

19 A time-space tradeoff for sorting on a general sequential model of computation
A. Borodin, S. Cook
April 1980 **Proceedings of the twelfth annual ACM symposium on Theory of computing STOC '80**
Publisher: ACM Press
Full text available: pdf(613.00 KB) Additional Information: full citation, abstract, references, citings, index terms

In a general sequential model of computation, no restrictions are placed on the way in which the computation may proceed, except parallel operations are not allowed. We show that in such an unrestricted environment TIME•SPACE&equil;&Ohgr;(N2/log N) in order to sort N elements, each in the range [1,N2].

20 Genetic algorithms: Crossover is provably essential for the Ising model on trees
Dirk Sudholt
June 2005 **Proceedings of the 2005 conference on Genetic and evolutionary computation GECCO '05**
Publisher: ACM Press
Full text available: pdf(193.03 KB) Additional Information: full citation, abstract, references, index terms

Due to experimental evidence it is incontestable that crossover is essential for some fitness functions. However, theoretical results without assumptions are difficult. So-called real royal road functions are known where crossover is proved to be essential, i.e., mutation-based algorithms have an exponential expected runtime while the expected runtime of a genetic algorithm is polynomially bounded. However, these functions are artificial and have been designed in such a way that crossover is ess ...

Keywords: expected optimization time, fitness sharing, ising model, mutation vs. crossover, theoretical analysis

Results 1 - 20 of 152 Result page: **1** 2 3 4 5 6 7 8 next

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+assert +compiler +variable SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **assert compiler variable**

Found **3,259** of **201,062**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale □ ▭ ▬ ▮ ■

1 A Compiler Analysis of Interprocedural Data Communication ■

Yonghua Ding, Zhiyuan Li

November 2003 **Proceedings of the 2003 ACM/IEEE conference on Supercomputing SC '03**

Publisher: IEEE Computer Society

Full text available: pdf(233.62 KB) Additional Information: full citation, abstract

This paper presents a compiler analysis for data communication for the purpose of transforming ordinary programs into ones that run on distributed systems. Such transformations have been used for process migration and computation offloading to improve the performance of mobile computing devices. In a client-server distributed environment, the efficiency of an application can be improved by careful partitioning of tasks between the server and the client. Optimal task partitioning depends on the t ...

2 High level programming for distributed computing ■

Jerome A. Feldman

June 1979 **Communications of the ACM**, Volume 22 Issue 6

Publisher: ACM Press

Full text available: pdf(1.78 MB) Additional Information: full citation, abstract, references, citings

Programming for distributed and other loosely coupled systems is a problem of growing interest. This paper describes an approach to distributed computing at the level of general purpose programming languages. Based on primitive notions of module, message, and transaction key, the methodology is shown to be independent of particular languages and machines. It appears to be useful for programming a wide range of tasks. This is part of an ambitious program of development in advanced programmin ...

Keywords: assertions, distributed computing, messages, modules

3 Gated SSA-based demand-driven symbolic analysis for parallelizing compilers ■

Peng Tu, David Padua

July 1995 **Proceedings of the 9th international conference on Supercomputing ICS '95**

Publisher: ACM Press

Full text available: pdf(1.08 MB) Additional Information: full citation, references, citings, index terms

4 Beyond induction variables: detecting and classifying sequences using a demand-driven SSA form ■

Michael P. Gerlek, Eric Stoltz, Michael Wolfe

January 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 1
Publisher: ACM Press

Full text available: pdf(2.27 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Linear induction variable detection is usually associated with the strength reduction optimization. For restructuring compilers, effective data dependence analysis requires that the compiler detect and accurately describe linear and nonlinear induction variables as well as more general sequences. In this article we present a practical technique for detecting a broader class of linear induction variables than is usually recognized, as well as several other sequence forms, including periodic, ...

Keywords: constant propagation, def-use chain, demand-driven, induction variable, static single assignment, strength reduction, wraparound variable

5 Constraint-based array dependence analysis
William Pugh, David Wonnacott
May 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 3
Publisher: ACM Press

Full text available: pdf(522.24 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Traditional array dependence analysis, which detects potential memory aliasing of array references is a key analysis technique for automatic parallelization. Recent studies of benchmark codes indicate that limitations of analysis cause many compilers to overlook large amounts of potential parallelism, and that exploiting this parallelism requires algorithms to answer new question about array references, not just get better answers to the old questions of aliasing. We need to ask about the ...

Keywords: Presburger Arithmetic, array dataflow analysis, dependence abstraction, dependence analysis, parallelization, static analysis

6 Early experiences with Euclid
David B. Wortman, James R. Cordy
March 1981 **Proceedings of the 5th international conference on Software engineering ICSE '81**
Publisher: IEEE Press

Full text available: pdf(540.80 KB) Additional Information: full citation, abstract, references, citings, index terms

The programming language Euclid was designed to be used for the construction of reliable and efficient systems software. This paper discusses the authors' experience in the design and implementation of the first large (about 60,000 source lines) piece of software written in Euclid. The emphasis in this paper is on how the various language features in Euclid affected the implementation of the software.

7 Experiences using the ParaScope Editor: an interactive parallel programming tool
Mary W. Hall, Timothy J. Harvey, Ken Kennedy, Nathaniel McIntosh, Kathryn S. McKinley, Jeffrey D. Oldham, Michael H. Paleczny, Gerald Roth
July 1993 **ACM SIGPLAN Notices , Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '93**, Volume 28 Issue 7
Publisher: ACM Press

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

The ParaScope Editor is an interactive parallel programming tool that assists knowledgeable users in developing scientific Fortran programs. It displays the results of sophisticated program analyses, provides a set of powerful interactive transformations, and supports program editing. This paper summarizes experiences of scientific programmers and tool designers using the ParaScope Editor. We evaluate existing features and describe enhancements in three key areas: user interface, analysis, ...

8 The Python compiler for CMU Common Lisp
Robert A. MacLachlan
January 1992 **ACM SIGPLAN Lisp Pointers , Proceedings of the 1992 ACM conference on LISP and functional programming LFP '92**, Volume V Issue 1
Publisher: ACM Press
Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

The Python compiler for CMU Common Lisp has been under development for over five years, and now forms the core of a production quality public domain Lisp implementation. Python synthesizes the good ideas from Lisp compilers and source transformation systems with mainstream optimization and retargetability techniques. Novel features include strict type checking and source-level debugging of compiled code. Unusual attention has been paid to the compiler's user interface.

9 High performance Fortran language specification
CORPORATE Rice University
December 1993 **ACM SIGPLAN Fortran Forum**, Volume 12 Issue 4
Publisher: ACM Press
Full text available: pdf(5.69 MB) Additional Information: full citation, abstract, citings, index terms

(PART I)Fortran Forum is reprinting this High Performance Fortran Language Specification over several issues. The current issue is devoted to the first four chapters of the HPFF Language Specification. Remaining chapters of the HPFF Language Specification, and the HPFF Journal of Development, will be printed in installments in filture issues of Fortran Forum.

10 Program checking
Graeme Williams
August 1979 **ACM SIGPLAN Notices , Proceedings of the 1979 SIGPLAN symposium on Compiler construction SIGPLAN '79**, Volume 14 Issue 8
Publisher: ACM Press
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references, index terms

In languages such as Pascal, the programmer can arrange to have the compiler check such things as the range of the value of a variable only by defining a new type or subtype. I have investigated how more powerful checking facilities might be provided if they were divorced from the type machinery, and also if the necessary language constructs were designed independent of what any particular compiler would check at compile-time. The first part of the project is the language design ...

11 Random testing of C calling conventions
Christian Lindig
September 2005 **Proceedings of the sixth international symposium on Automated analysis-driven debugging AADEBUG'05**
Publisher: ACM Press
Full text available: pdf(199.71 KB) Additional Information: full citation, abstract, references, citings, index terms

In a C compiler, function calls are difficult to implement correctly because they must respect a platform-specific calling convention. But they are governed by a simple invariant: parameters passed to a function must be received unaltered. A violation of this invariant signals an inconsistency in a compiler. We automatically test the consistency of C compilers using randomly generated programs. An inconsistency manifests itself as an assertion failure when compiling and running the generated cod ...

Keywords: C, calling convention, compiler, composition, consistency, random testing

12 The Pascal dynamic array controversy and a method for enforcing global assertions
Michael N. Condict
November 1977 **ACM SIGPLAN Notices**, Volume 12 Issue 11
Publisher: ACM Press
Full text available: pdf(321.76 KB) Additional Information: full citation, references

13 Self-assessment procedure VIII: a self-assessment procedure dealing with the programming language Ada
Peter Wegner
October 1981 **Communications of the ACM**, Volume 24 Issue 10
Publisher: ACM Press
Full text available: pdf(2.41 MB) Additional Information: full citation, references, citings, index terms

14 Tera hardware-software cooperation
Gail Alverson, Preston Briggs, Susan Coatney, Simon Kahan, Richard Korry
November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM) Supercomputing '97**
Publisher: ACM Press
Full text available: pdf(217.50 KB) Additional Information: full citation, abstract, references, citings

The development of Tera's MTA system was unusual. It respected the need for fast hardware and large shared memory, facilitating execution of the most demanding parallel application programs. But at the same time, it met the need for a clean machine model enabling calculated compiler optimizations and easy programming; and the need for novel architectural features necessary to support fast parallel system software. From its inception, system and application needs have molded the MTA architecture. ...

15 Eliminating false data dependences using the Omega test
William Pugh, David Wonnacott
July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation PLDI '92**, Volume 27 Issue 7
Publisher: ACM Press
Full text available: pdf(1.23 MB) Additional Information: full citation, abstract, references, citings, index terms

Array data dependence analysis methods currently in use generate false dependences that can prevent useful program transformations. These false dependences arise because the questions asked are conservative approximations to the questions we really should be asking. Unfortunately, the questions we really should be asking go beyond integer programming and require decision procedures for a sublcass of Presburger formulas. In this paper, we describe how to extend the Omega test so that it can ...

16 Type-safe linkage for variables and functions
Diomidis Spinellis
August 1991 **ACM SIGPLAN Notices**, Volume 26 Issue 8
Publisher: ACM Press
Full text available: pdf(360.16 KB) Additional Information: full citation, abstract, index terms

In a separate compilation environment type checks across modules are difficult to implement, because the natural place to perform them, the linker, is rarely under the control of the compiler developer. A solution to this problem, presented in the C++ Reference Manual, does not cope with global variables and function return types. It is asserted that lifting those limitations would require modifying the linker or providing an environment for separate compilation. We present a solution that lifts ...

17 An approach to compiler correctness
Laurian M. Chirica, David F. Martin
April 1975 **ACM SIGPLAN Notices , Proceedings of the international conference on Reliable software**, Volume 10 Issue 6
Publisher: ACM Press
Full text available: pdf(611.43 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper is a preliminary report on an experiment in applying Floyd's method of inductive assertions to the compiler correctness problem. Practical postfix translators are considered, and the semantics of source and object languages are characterized by Floyd verification conditions. Compiler correctness proofs are partitioned into two parts. The first part deals with proofs of the syntactic and translational phase of compilation, and generates semantic equivalence theorems which are prov ...

Keywords: Compiler correctness, Inductive assertions, Semantic equivalence, Syntax-directed translation

18 Functional back-ends within the lambda-sigma calculus
Thérèse Hardin, Luc Maranget, Bruno Pagano
June 1996 **ACM SIGPLAN Notices , Proceedings of the first ACM SIGPLAN international conference on Functional programming ICFP '96**, Volume 31 Issue 6
Publisher: ACM Press
Full text available: pdf(904.44 KB) Additional Information: full citation, abstract, references, citings, index terms

We define a weak &lambda;-calculus, &lambda;&sigma;w, as a subsystem of the full &lambda;-calculus with explicit substitutions &lambda;&sigma;&uArr;. We claim that &lambda;&sigma;w could be the archetypal output language of functional compilers, just as the &lambda;-calculus is their universal input language. Furthermore, &lambda;&sigma;&uArr; could be the adequate theory to establish the correctness of simplified function ...

19 Breaking the complexity barrier again
Terry Winograd
November 1973 **ACM SIGIR Forum , ACM SIGPLAN Notices , Proceedings of the 1973 meeting on Programming languages and information retrieval SIGPLAN '73**, Volume 9 , 10 Issue 3 , 1
Publisher: ACM Press
Full text available: pdf(1.75 MB) Additional Information: full citation, references, citings

20 Notes on the design of Euclid
G. J. Popek, J. J. Horning, B. W. Lampson, J. G. Mitchell, R. L. London
March 1977 **ACM SIGPLAN Notices , ACM SIGSOFT Software Engineering Notes , ACM SIGOPS Operating Systems Review , Proceedings of an ACM conference on Language design for reliable software**, Volume 12 , 2 , 11 Issue 3 , 2 , 2
Publisher: ACM Press
Full text available: pdf(903.58 KB) Additional Information: full citation, abstract, citings, index terms

Euclid is a language for writing system programs that are to be verified. We believe that verification and reliability are closely related, because if it is hard to reason about programs using a language feature, it will be difficult to write programs that use it properly. This paper discusses a number of issues in the design of Euclid, including such topics as the scope of names, aliasing, modules, type-checking, and the confinement of machine dependencies; it gives some of the reasons for ...

Keywords: Aliasing, Data encapsulation, Euclid, Legality assertions, Machine dependencies, Parameterized types, Pascal, Reliability, Storage allocation, Systems programming language, Verification, Visibility of names

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

IEEE Xplore® RELEASE 2.3

Home | Login | Logout | Access Information | Alerts | Sitemap | Help

Welcome United States Patent and Trademark Office

□ Search Results

BROWSE SEARCH IEEE XPLORE GUIDE SUPPORT

Results for **"((assert and compiler)<in>metadata)"**
Your search matched **7** of **1566306** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail printer friendly

» Search Options

View Session History

New Search

Modify Search

((assert and compiler)<in>metadata) Search >

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

view selected items **Select All** **Deselect All**

□ 1. **Software testing concept used for MAS/C-compiler**
Popovic, M.; Kovacevic, V.; Temerinac, M.;
Euromicro Conference, 2000. Proceedings of the 26th
Volume 2, 5-7 Sept. 2000 Page(s):224 - 229 vol.2
Digital Object Identifier 10.1109/EURMIC.2000.874422
AbstractPlus | Full Text: PDF(744 KB) **IEEE CNF**
Rights and Permissions

□ 2. **Memory Protection through Dynamic Access Control**
Kun Zhang; Tao Zhang; Santosh Pande;
Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on
Dec. 2006 Page(s):123 - 134
Digital Object Identifier 10.1109/MICRO.2006.33
AbstractPlus | Full Text: PDF(268 KB) **IEEE CNF**
Rights and Permissions

□ 3. **An assessment of the MODSIM/TWOS parallel simulation environment**
Rich, D.O.; Michelsen, R.E.;
Simulation Conference, 1991. Proceedings., Winter
8-11 Dec. 1991 Page(s):509 - 518
Digital Object Identifier 10.1109/WSC.1991.185653
AbstractPlus | Full Text: PDF(896 KB) **IEEE CNF**
Rights and Permissions

□ 4. **Computational improvements in Prolog applications by predicate variable pointers**
Karam, G.M.;
Software Engineering, IEEE Transactions on
Volume 16, Issue 5, May 1990 Page(s):490 - 497
Digital Object Identifier 10.1109/32.52772
AbstractPlus | Full Text: PDF(656 KB) **IEEE JNL**
Rights and Permissions

□ 5. **Compiling SIMULA: a historical study of technological genesis**
Holmevik, J.R.;
Annals of the History of Computing, IEEE
Volume 16, Issue 4, Winter 1994 Page(s):25 - 37
Digital Object Identifier 10.1109/85.329756
AbstractPlus | Full Text: PDF(1248 KB) **IEEE JNL**
Rights and Permissions

□ 6. **A practical architecture for reliable quantum computers**

Oskin, M.; Chong, F.T.; Chuang, I.L.;
Computer
Volume 35, Issue 1, Jan. 2002 Page(s):79 - 87
Digital Object Identifier 10.1109/2.976922

AbstractPlus | References | Full Text: PDF(170 KB) **IEEE JNL**
Rights and Permissions

□ 7. **Deconstructing commit**
Bell, G.B.; Lipasti, M.H.;
Performance Analysis of Systems and Software, 2004 IEEE International Symposium on - ISPASS
2004 Page(s):68 - 77
Digital Object Identifier 10.1109/ISPASS.2004.1291357

AbstractPlus | Full Text: PDF(1924 KB) **IEEE CNF**
Rights and Permissions

Help Contact Us Privacy & Security IEEE.org

Indexed by Inspec®

Home | Login | Logout | Access Information | Alerts | Sitemap | Help

Welcome United States Patent and Trademark Office

□ Search Results

BROWSE SEARCH IEEE XPLORE GUIDE SUPPORT

Results for **"(assert and variable and use<in>metadata)"**
Your search matched **3405** of **1566306** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail printer friendly

» Search Options

View Session History

New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

Modify Search

(assert and variable and use<in>metadata) Search

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

view selected items **Select All** **Deselect All**

View: **1-25** | 26-50 | 51-75 | 76-100

□ 1. **Synthesis of arithmetic hardware using hardware metafunctions**
Chin, S.-K.; Stabler, E.P.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 9, Issue 8, Aug. 1990 Page(s):793 - 803
Digital Object Identifier 10.1109/43.57787
AbstractPlus | Full Text: PDF(904 KB) **IEEE JNL**
Rights and Permissions

□ 2. **Formal verification of concurrent programs using the Larch prover**
Chetali, B.;
Software Engineering, IEEE Transactions on
Volume 24, Issue 1, Jan. 1998 Page(s):46 - 62
Digital Object Identifier 10.1109/32.663997
AbstractPlus | References | Full Text: PDF(248 KB) **IEEE JNL**
Rights and Permissions

□ 3. **A simple method for extracting models from protocol code**
Lie, D.; Chou, A.; Engler, D.; Dill, D.L.;
Computer Architecture, 2001. Proceedings. 28th Annual International Symposium on
30 June-4 July 2001 Page(s):192 - 203
Digital Object Identifier 10.1109/ISCA.2001.937448
AbstractPlus | Full Text: PDF(176 KB) **IEEE CNF**
Rights and Permissions

□ 4. **Debugging Larch shared language specifications**
Garland, S.J.; Guttag, J.V.; Horning, J.J.;
Software Engineering, IEEE Transactions on
Volume 16, Issue 9, Sept. 1990 Page(s):1044 - 1057
Digital Object Identifier 10.1109/32.58789
AbstractPlus | Full Text: PDF(1176 KB) **IEEE JNL**
Rights and Permissions

□ 5. **A unified approach to the synthesis of fully testable sequential machines**
Devadas, S.; Keutzer, K.W.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 10, Issue 1, Jan. 1991 Page(s):39 - 50
Digital Object Identifier 10.1109/43.62790
AbstractPlus | Full Text: PDF(1172 KB) **IEEE JNL**
Rights and Permissions

□ 6. **A practical approach to programming with assertions**

Rosenblum, D.S.;
Software Engineering, IEEE Transactions on
Volume 21, Issue 1, Jan. 1995 Page(s):19 - 31
Digital Object Identifier 10.1109/32.341844

AbstractPlus | References | Full Text: PDF(1200 KB) **IEEE JNL**
Rights and Permissions

□ 7. **A CMOS combinational circuit-design method using mixed logic concepts**
Hudson, W.B.; Beasley, J.S.; Steelman, J.E.;
Education, IEEE Transactions on
Volume 38, Issue 3, Aug. 1995 Page(s):266 - 273
Digital Object Identifier 10.1109/13.406505

AbstractPlus | Full Text: PDF(532 KB) **IEEE JNL**
Rights and Permissions

□ 8. **Automatically generating malicious disks using symbolic execution**
Junfeng Yang; Can Sar; Twohey, P.; Cadar, C.; Engler, D.;
Security and Privacy, 2006 IEEE Symposium on
21-24 May 2006 Page(s):15 pp.
Digital Object Identifier 10.1109/SP.2006.7

AbstractPlus | Full Text: PDF(488 KB) **IEEE CNF**
Rights and Permissions

□ 9. **Enhancing security using legality assertions**
Lei Wang; Cordy, J.R.; Dean, T.R.;
Reverse Engineering, 12th Working Conference on
7-11 Nov. 2005 Page(s):10 pp.
Digital Object Identifier 10.1109/WCRE.2005.36

AbstractPlus | Full Text: PDF(216 KB) **IEEE CNF**
Rights and Permissions

□ 10. **An observationally complete program logic for imperative higher-order functions**
Honda, K.; Yoshida, N.; Berger, M.;
Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on
26-29 June 2005 Page(s):270 - 279
Digital Object Identifier 10.1109/LICS.2005.5

AbstractPlus | Full Text: PDF(368 KB) **IEEE CNF**
Rights and Permissions

□ 11. **Verifying Web applications using bounded model checking**
Yao-Wen Huang; Fang Yu; Hang, C.; Chung-Hung Tsai; Lee, D.T.; Sy-Yen Kuo;
Dependable Systems and Networks, 2004 International Conference on
28 June-1 July 2004 Page(s):199 - 208
Digital Object Identifier 10.1109/DSN.2004.1311890

AbstractPlus | Full Text: PDF(358 KB) **IEEE CNF**
Rights and Permissions

□ 12. **Testing production system programs**
Antoniou, G.; Jack, O.;
Software Reliability Engineering, 1998. Proceedings. The Ninth International Symposium on
4-7 Nov. 1998 Page(s):214 - 221
Digital Object Identifier 10.1109/ISSRE.1998.730884

AbstractPlus | Full Text: PDF(404 KB) **IEEE CNF**
Rights and Permissions

□ 13. **Formal verification of pipeline control using controlled token nets and abstract interpretation**
Pei-Hsin Ho; Isles, A.J.; Kam, T.;
Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers. 1998 IEEE/ACM International Conference on
8-12 Nov 1998 Page(s):529 - 536

AbstractPlus | Full Text: PDF(872 KB) **IEEE CNF**
Rights and Permissions

□ **14. HSIS: A BDD-Based Environment for Formal Verification**
Aziz, A.; Balarin, F.; Cheng, S.-T.; Hojati, R.; Kam, T.; Krishnan, S.C.; Ranjan, R.K.; Shiple, T.R.; Singhal, V.; Tasiran, S.; Wang, H.-Y.; Brayton, R.K.; Sangiovanni-Vincentelli, A.L.;
Design Automation, 1994. 31st Conference on
6-10 June 1994 Page(s):454 - 459

AbstractPlus | Full Text: PDF(89 KB) **IEEE CNF**
Rights and Permissions

□ **15. Automatic checking of aggregation abstractions through state enumeration**
Seungjoon Park; Das, S.; Dill, D.L.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 19, Issue 10, Oct. 2000 Page(s):1202 - 1210
Digital Object Identifier 10.1109/43.875327

AbstractPlus | References | Full Text: PDF(176 KB) **IEEE JNL**
Rights and Permissions

□ **16. Ontology management for large-scale e-commerce applications**
Lee, J.; Goodwin, R.;
Data Engineering Issues in E-Commerce, 2005. Proceedings. International Workshop on
9 April 2005 Page(s):7 - 15
Digital Object Identifier 10.1109/DEEC.2005.14

AbstractPlus | Full Text: PDF(1504 KB) **IEEE CNF**
Rights and Permissions

□ **17. The Architecture of Coherent Information System: A General Problem Solving System**
Srinivasan, C.V.;
Computers, IEEE Transactions on
Volume C-25, Issue 4, Apr 1976 Page(s):390 - 402

AbstractPlus | Full Text: PDF(3456 KB) **IEEE JNL**
Rights and Permissions

□ **18. Direct stability analysis of electric power systems using energy functions: theory, applications, and perspective**
Hsiao-Dong Chang; Chia-Chi Chu; Cauley, G.;
Proceedings of the IEEE
Volume 83, Issue 11, Nov. 1995 Page(s):1497 - 1529
Digital Object Identifier 10.1109/5.481632

AbstractPlus | Full Text: PDF(3108 KB) **IEEE JNL**
Rights and Permissions

□ **19. MALLET - a multi-agent logic language for encoding teamwork**
Xiaocong Fan; Yen, J.; Miller, M.; Ioerger, T.R.; Volz, R.;
Knowledge and Data Engineering, IEEE Transactions on
Volume 18, Issue 1, Jan. 2006 Page(s):123 - 138
Digital Object Identifier 10.1109/TKDE.2006.13

AbstractPlus | Full Text: PDF(904 KB) **IEEE JNL**
Rights and Permissions

□ **20. A language for construction of belief networks**
Goldman, R.P.; Charniak, E.;
Pattern Analysis and Machine Intelligence, IEEE Transactions on
Volume 15, Issue 3, March 1993 Page(s):196 - 208
Digital Object Identifier 10.1109/34.204902

AbstractPlus | Full Text: PDF(1232 KB) **IEEE JNL**
Rights and Permissions

□ **21. Modular verification of data abstractions with shared realizations**
Ernst, G.W.; Hookway, R.J.; Ogden, W.F.;
Software Engineering, IEEE Transactions on
Volume 20, Issue 4, April 1994 Page(s):288 - 307
Digital Object Identifier 10.1109/32.277576

AbstractPlus | Full Text: PDF(2120 KB) **IEEE JNL**
Rights and Permissions

□ **22. Don't cares in synthesis: theoretical pitfalls and practical solutions**
Brand, D.; Bergamaschi, R.A.; Stok, L.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 17, Issue 4, April 1998 Page(s):285 - 304
Digital Object Identifier 10.1109/43.703819

AbstractPlus | References | Full Text: PDF(396 KB) **IEEE JNL**
Rights and Permissions

□ **23. A unified symbolic evaluation framework for parallelizing compilers**
Fahringer, T.; Scholz, B.;
Parallel and Distributed Systems, IEEE Transactions on
Volume 11, Issue 11, Nov. 2000 Page(s):1105 - 1125
Digital Object Identifier 10.1109/71.888633

AbstractPlus | References | Full Text: PDF(464 KB) **IEEE JNL**
Rights and Permissions

□ **24. Robust Boolean reasoning for equivalence checking and functional property verification**
Kuehlmann, A.; Paruthi, V.; Krohm, F.; Ganai, M.K.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 21, Issue 12, Dec. 2002 Page(s):1377 - 1394
Digital Object Identifier 10.1109/TCAD.2002.804386

AbstractPlus | References | Full Text: PDF(1274 KB) **IEEE JNL**
Rights and Permissions

□ **25. Automatic discovery of API-level exploits**
Bryant, R.E.; Jha, S.; Reps, T.W.; Seshia, S.A.; Ganapathy, V.;
Software Engineering, 2005. ICSE '05. Proceedings of the 27th International Conference on
15-21 May 2005 Page(s):312 - 321

AbstractPlus | Full Text: PDF(512 KB) **IEEE CNF**
Rights and Permissions

View: 1-25 | 26-50 | 51-75 | 76-100

Help Contact Us Privacy & Security IEEE.org

Indexed by
Inspec®

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	83	parallel with computer with compiler and (factor$4 (fud))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:49
L3	148	(enter$4 insert$4 input$4) with assert with (user compiler)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 16:12
L16	8	variable with use same java same compiler	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:25
L18	13	(variable with use) and (assert with statement)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 16:11
L21	12	((fud) (factored adj use$def)) and assert	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:54
L22	22	((fud) (use$def) (use adj def) (usage adj definition)) and assert	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:33
L24	59	(identif$4 find$4 determin$4 discover$4) with (variable adj use)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:54
L26	13	(identif$4 find$4 determin$4 discover$4) with (variable adj use) and compil$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:42

L29	10	((dom) (denominator adj tree)) with (control adj flow)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:47
L30	256	((dom) (denominator adj tree)) and assert	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:55
L31	8	((dom) (denominator adj tree)) same assert	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:48
L41	19	parallel with computer with compiler and (717/116 717/140 717/148 717/154 717/155)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:57
L42	1	(identif$4 find$4 determin$4 discover$4) with (variable adj use) and (717/116 717/140 717/148 717/154 717/155)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 15:58
L43	1	(identif$4 find$4 determin$4 discover$4) with (variable adj use) and (717/116 717/140 717/148 717/154 717/155)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 16:04
L44	1	assert with statement same variable and (717/116 717/140 717/148 717/154 717/155)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/08 16:04
L45	54	(enter$4 insert$4 input$4) with assert with (user compiler)	US-PGPUB	OR	ON	2007/05/08 16:11
L46	14	assert same variable same (use usage)	US-PGPUB	OR	ON	2007/05/08 16:10
L49	1	(enter$4 insert$4 input$4) with assert with (user compiler).clm.	US-PGPUB	OR	ON	2007/05/08 16:11
L52	1	(enter$4 insert$4 input$4) with assert with (user compiler).clm.	US-PGPUB	OR	ON	2007/05/08 16:12

L53	39	compil$4 same (variable with information).clm.	US-PGPUB	OR	ON	2007/05/08 16:12
L56	4	((dom) (denominator adj tree)) same compil$4.clm.	US-PGPUB	OR	ON	2007/05/08 16:13